10-27-00

A

10/2000 FORM 1

TRANSMITTAL FORM FOR FILING PATENT APPLICATION

JC860 U.S. PTO 10/26/00

Attorney
Docket No.: SYCMR-031XX

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
Ten Post Office Square
Boston, Massachusetts 02109
Telephone: (617) 542-2290
Telecopier: (617) 451-0313

JC917 U.S. PTO 09/696860 10/26/00

Express Mail No: EL418427798US
Date: October 26, 2000

BOX PATENT APPLICATION
Assistant Commissioner for Patents
Washington, D.C. 20231

First Named Inventor or Application Identifier: Chikong Shue et al.

Sir:

Transmitted herewith under 37 CFR § 1.53 for filing is the patent application of:

Inventor: Chikong Shue, John Thomas Moy, Tao Yang, Raymond Xie

Entitled: NETWORK EMULATOR

[ ] This is a request for filing a [ ] **continuation** [ ] **divisional** [ ] **continuation in-part** application under §1.53(b) of prior Application No._____, filed _____ entitled:

Enclosed are:

[X] 26 pages of written description, claims and Abstract, inclusive

[X] 4 sheets of [ ] informal [X] formal drawings of Figs. 1-6 (one set)

[X] Oath or Declaration
- [ ] Newly executed (original or copy)
- [ ] Copy from prior application (37 CFR 1.63(d)) (for continuation/divisional). The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.
- [X] (Unsigned) To be filed later

[ ] Cover sheet and Assignment of the invention to:

[ ] Certified copy of a _____ application (if foreign priority is claimed) with letter claiming priority under Rule 55.

[ ] Information Disclosure Statement with __ citations

[ ] Preliminary amendment is enclosed.

[X] Return receipt postcard

[ ] Other:

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

Attorney Docket No.: SYCMR-031XX

TRANSMITTAL FORM FOR FILING PATENT APPLICATION (CONTINUED)

[ ] Verified statement of Small Entity status (§1.9 and §1.27)

[ ] Verified statement of Small Entity was filed in prior application. Status still proper and desired

[ ] Priority is claimed under 35 USC § 120 as indicated on the attached sheet 4.

[ ] Priority is claimed under 35 USC §119(a)-(d) as indicated on the attached sheet 4.

[ ] Priority is claimed under 35 USC §119 (e) as indicated on the attached sheet 4.

[X] David A. Dagg is hereby appointed Associate Attorney by:
Registration No.: 37,809

Attorney of Record Victor B. Lebovici
Registration No.:30,864

[ ] **Power of Attorney** in the originally-filed application has been granted to one or more of the registered attorneys listed below. The attorneys listed below not previously granted power in the originally-filed application, as well as _____, are hereby given associate power:
Registration No.:

Stanley M. Schurgin, Reg. No. 20,979
Charles L. Gagnebin III, Reg. No. 25,467
Paul J. Hayes, Reg. No. 28,307
Victor B. Lebovici, Reg. No. 30,864
Eugene A. Feher, Reg. No. 33,171
Beverly E. Hjorth, Reg. No. 32,033
Holliday C. Heine, Reg. No. 34,346
Gordon R. Moriarty, Reg. No. 38,973

[ ] Cancel in this application original claims _____of the prior application before calculating the filing fee.

[ ] Add in this application claims _____ per amendment before calculating fee.

CLAIMS FILED:	MINUS BASE:	EXTRA CLAIMS:	RATE:	BASIC FEE:
				$710.00
Independent	2 - 3	=	x $80.00 =	$0.00
Total	21 - 20	= 1	x $18.00 =	$18.00
[ ] Multiple Dependent Claims (1st presentation)			+ $270.00 =	$0.00
			SUBTOTAL FILING FEE	$728.00
Small Entity filing, divide by 2. (Note: verified statement must be attached per §1.9, §1.27, §1.28.)				$0.00
			TOTAL Filing Fee	$728.00

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

Attorney Docket No.:SYCMR-031XX

TRANSMITTAL FOR FILING PATENT APPLICATION (CONTINUED)

[ ] The filing fee has been calculated above; a check in the amount of _____ is enclosed.

[X] The filing fee will be submitted at a later date.

[X] In the event a Petition for Extension of Time under 37 CFR §1.17 is required by this paper and not otherwise provided, such Petition is hereby made and authorization is provided herewith to charge Deposit Account No. 23-0804 for the cost of such extension.

[ ] The Commissioner is hereby authorized to charge payment of any additional filing fees under 37 CFR §1.16 associated with this communication or credit any overpayment to Deposit Account No. 23-0804.

[X] Customer Number 207

Address all future communications to:

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
Ten Post Office Square
Boston, Massachusetts 02109
Telephone: (617) 542-2290
Telecopier: (617) 451-0313

Attorney of Record: David A. Dagg
Registration No. 37,809

Attorney Docket No.: SYCMR-031XX

TRANSMITTAL FOR FILING PATENT APPLICATION (CONTINUED)

[ ] Priority is claimed under 35 USC § 120 of prior Application(s) No. _____, filed_____, entitled:

[ ] The above-identified application(s) is/are assigned of record to:

[ ] Priority is claimed under 35 USC § 119 (a)-(d) of the following application(s).

(Application Number)	(Country)	(Filing Date)

[ ] The above-identified application(s) is/are assigned of record to:

[ ] Priority is claimed under 35 USC § 119 (e) of the following provisional application(s).

(Application Number)	(Filing Date)

[ ] The above-identified provisional application(s) is/are assigned of record to:

[ ] The claim of small entity status in the above-identified provisional application(s) is made in this application and a copy of the small entity form(s) from the provisional application(s) is/are enclosed.

SUBMIT IN TRIPLICATE

DAD/kdm
236458

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

TITLE OF THE INVENTION

NETWORK EMULATOR

CROSS REFERENCE TO RELATED APPLICATIONS

N/A

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

N/A

BACKGROUND OF THE INVENTION

The present invention relates generally to simulating a computer network, and more specifically to a system and method for large scale testing and validation of optical switching equipment and software.

As the need for data and voice communications bandwidth has steadily increased, communication networks have grown to include ever larger numbers of sophisticated networking devices, such as optical switches. These networking devices often include complex data forwarding functionality, which is typically provided using a combination of software and hardware components. For example, networking devices often include multiple, interconnected, printed circuit boards, sometimes referred to as "blades" or "cards", one or more of which may include microprocessor-based subsystems for execution of various software components. The operation of such networking devices is significantly dependent on the

ATTORNEY DOCKET NO. SYCMR-031XX
WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES, LLP
TEL. (617) 542-2290
FAX. (617) 451-0313

Express Mail Numb
EL418427798US

specific network configuration in which they are deployed. Accordingly, testing of such devices should include tests of their operation in a variety of network configurations.

For example, during development of software for execution within a networking device, software verification testing should test the software in a variety of network configurations. Because the number of networking devices required to perform a comprehensive set of such tests may be relatively large, and because the amount of time necessary to reconfigure actual networking devices into the various specific configurations to be tested may be so great as to be impractical, testing based on emulation of various network configurations may sometimes be desirable. In fact, a software emulation system capable of emulating large and complex networks would significantly reduce overall software development time and equipment expense.

Additionally, it may be desirable to test proposed or hypothetical network configurations when designing a new network, or when planning modifications to an existing network. In this regard, customers of networking device providers often wish to determine how a specific configuration of devices will operate before purchasing the actual equipment. Network emulation may thus advantageously be used during network planning and design, and to perform "what if" testing of possible network modifications.

Various existing systems have attempted to provide effective software emulation of communication networks. Some such existing systems have emulated operation of networking device software using software that has been "ported" from an execution environment present on the networking devices, to another execution environment used

within the emulator system. For example, the execution environment within a networking device may include a microprocessor, operating system, and/or other aspects which differ from those used in the emulation system. In order for the networking device software to execute within the emulation environment, it has, in such existing systems, been modified or "ported" from the form in which it is used within the networking device. However, such code modifications raise issues of whether the modified code behaves the same as the actual code that is or will be deployed in the networking device. Accordingly, the accuracy of testing provided by such existing systems may be poor.

For the reasons stated above, it would be desirable to have a network emulation system that supports testing of large numbers of complex networking devices using code that is not modified from the form in which it would be used within the actual networking devices.

BRIEF SUMMARY OF THE INVENTION

In accordance with the present invention, a system and method for providing an emulated network including multiple emulated networking devices are disclosed. In the disclosed system, a number of network node executable images are employed. Each of the network node executable images corresponds to an emulated networking device. The emulated networking devices represent physical networking devices, such as optical switches. Each of the network node executable images includes executable code that is operable to execute without modification on a physical networking

device. In a preferred embodiment, the executable code that is operable to execute without modification on a physical networking device includes executable code for signaling, routing, trunk management, circuit management and port management. Further in the preferred embodiment, the executable code that is operable to execute without modification on a corresponding physical network device is executable code that is operable to be executed within one of several cards that make up the corresponding physical network device. Additionally, the executable code that is operable without modification on a corresponding physical network device may further be linked to other executable code within each network node executable image which may, for example, not be operable without modification on the corresponding physical network device. Such modified code, for example, may perform other functions within the corresponding physical network device, and be executed on other cards within the physical network device. Such modified code may further be generated from a common source code program as is used to generate actual code operable for execution on such other cards within the corresponding physical network device.

More specifically, in a further aspect of the disclosed system, the executable code of the network node executable image that is operable without modification on a corresponding physical network device may include a topology related program component for maintaining a representation of the topology of the emulated network, a trunk manager program component for maintaining at least one emulated trunk between the corresponding emulated networking device and an adjacent other emulated networking device, and a

circuit manager program component operable to maintain at least one emulated circuit between the corresponding emulated networking device and another one of the emulated networking devices.

Various numbers of network node executable images may be executed on a single emulator machine. For example, in a first illustrative embodiment, multiple network node executable images may execute on a single emulator machine. Such a configuration is efficient in terms of emulator machine resources. Alternatively, each of the network node executable images may execute on separate, respective emulator machines. This type of configuration may provide highly accurate performance testing. Moreover, multiple emulator machines may be used to execute a single one of the network node executable images. Such a configuration would be useful, for example, when testing communication functions between components of a network node executable image that are assigned to different cards within the emulated networking device. In the case where multiple emulator machines are employed, they may be interconnected by a local area network such as an Ethernet LAN. Moreover, each of the plurality of network node executable images further includes a code component used to provide communication during emulator operation, and an alternative code component used during operation within an actual physical networking device. In an illustrative embodiment, these alternative code components are provided in two network layer drivers. The appropriate network layer driver is loaded for execution based on a configuration file which indicates whether the specific network node executable image is being used within

the network emulator, or within an actual physical networking device.

In this way there is disclosed a system and method for emulating a number of networking devices within a communications network. The disclosed system provides validation of operation within various network topologies, since the executable code representing each of the emulated networking devices is the same executable image that is used in the actual physical devices being emulated. The disclosed system may be used to accurately test the operation of new software images prior to installation of the images within the physical networking devices on which they are to be used, and for testing of potential network designs and modifications.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING

The invention will be more fully understood by reference to the following detailed description of the invention in conjunction with the drawings, of which:

Fig. 1 shows a number of network node executable images executing within a single emulator machine;

Fig. 2 shows a single network node executable image executing within a single emulator machine;

Fig. 3 shows executable images associated with individual component cards of a networking device executing within respective emulator machines;

Fig. 4 shows the functional components within a network node executable image;

Fig. 5 illustrates operation of network node executable images within the emulator system and within a physical network; and

Fig. 6 shows an example of a networking device, such as an optical switch, including a number of separate printed circuit boards ("cards") installed within a backplane.

DETAILED DESCRIPTION OF THE INVENTION

As illustrated in Fig. 1, an emulator machine 10 may be used to execute a number of network node executable images 12, in conjunction with an operating system 14. The emulator machines shown in Figs. 1-3 may be embodied using any appropriate code execution platform, such as a personal computer, workstation, or other hardware system. An emulator machine may include one or more processors 15 and associated memory 13 for storing code to be executed using such processors. The operating system 14 may be embodied using, for example, a conventional operating system such as those provided by Microsoft Corporation (Windows™, Windows NT™), UNIX or a UNIX-like operating system, such as Linux, or any other appropriate operating system for a particular implementation.

The network node executable images 12 represent code that is directly executable on the underlying hardware of the emulator machine 10. In an alternative embodiment, the network node executable images may be indirectly executable on the emulator machine 10 through use of a virtual machine (not shown).

As shown in Fig. 1, the disclosed system may be embodied or configured such that several network node executable images 12 are executed within a single emulator machine 10. Such a configuration may be efficient in terms

of emulating network configurations involving multiple nodes using a single emulator machine 10. Other configurations are described in connection with the following Figs. 2-3.

Fig. 2 illustrates a configuration of the disclosed system in which a single emulator machine 20 is used to execute a single network node executable image 22, in conjunction with an operating system 24. The emulator machine 20 of Fig. 2 is further shown including a processor 25 and associated memory 23. The configuration of Fig. 2 may be useful for testing the operation of a single network node executable image, for example, in the case where new code has been developed and should be tested for correctness. Further, in an illustrative embodiment, a number of emulator machines, such as the emulator machine 20, each having a single network node executable image executing thereon, may be interconnected through a network such as a local area network. Such an interconnection of multiple emulator machines having single network node executable images executing thereon may be used to accurately emulate the behavior and/or performance of the same number of physical networking devices within an actual network.

Referring now to Fig. 3, a local area network 40 is shown interconnecting the emulator machines 32, 34, 36 and 38. As shown in Fig. 3, the disclosed system may be used to emulate the behavior of individual cards that provide a portion of the functionality within actual physical networking devices. As shown in Fig. 3, the emulator machine 32 includes a network node executable image 42 for a card A stored with an operating system 44 in a memory 43, for execution on a processor 45. The emulator machine 34

includes a network node executable image 46 for a card B stored with an operating system 48 in a memory 46, for execution on a processor 49. The emulator machine 36 includes a network node executable image 50 for a card C stored with an operating system 52 in a memory 51, for execution on a processor 53. Emulator machine 38 includes a network node executable image 54 stored with an operating system 56 in a memory 55, for execution on a processor 57. The network node executable images shown in Fig. 3 may correspond to cards from within a single or multiple actual physical networking devices. For example, card A, card B, card C and card D may be identical cards of different networking devices within the emulated network. In an illustrative embodiment, card A, card B, card C and card D are each Switch Management Cards (SMCs) for respective ones of individual networking devices within an emulated network. Such SMCs may, for example, provide functionality related to signaling, routing, trunk management, circuit management, and port management. Alternatively, card A, card B, card C and card D may be separate, dissimilar cards within a single emulated device.

Fig. 4 shows an illustrative embodiment of a network node executable image 70, including signaling and routing functionality 72, as well as trunk, circuit and port manager functionality 74, and platform functionality 76. As it is generally known, signaling within the context of telephony related systems generally involves the exchange of information between nodes within a network in order to set up, control, and terminate a number of calls. Two forms of signaling may be provided by one or more signaling code entities within the signaling and routing functionality 72

of the network node executable image 70. First, "in-band" signaling may be provided, in which signaling information is exchanged within the same channel as the subject call or calls. Second, "out-of-band" signaling may be performed on separate, dedicated signaling channels. The signaling provided through the signaling and routing functionality 72 of the network node executable image 70 may include either in-band signaling, out-of-band signaling, or both.

Further for purposes of illustration, a routing code entity provided within the signaling and routing functionality 72 may, for example, implement the OSPF (Open Shortest Path First) router protocol. As it is generally known, OSPF is a router protocol which may be used as an alternative to the Routing Information Protocol (RIP). Like RIP, OSPF is designated by the Internet Engineering Task Force (IETF) as one of several Interior Gateway Protocols (IGPs). In an alternative embodiment, the routing program code provided within the signaling and routing functionality 72 may be based on RIP.

An illustrative embodiment of a trunk manager program code entity within the trunk, circuit and port manager functionality 74 is now described. As used herein, the term "trunk" shall refer to a logical link between two networking devices over which one or more circuits can be provisioned. The trunk manager code entity generally provides the following functions:

(a) Management and maintenance of state information for all trunks within the networking device, including creation and/or deletion of trunks in response to commands or instructions received, for example, from one or more

network management code entities. The trunk manager is further responsible for maintaining the integrity of a trunk database after initialization or recovery from a failure.

(b) Bandwidth reservation and port mapping services.

(c) Translation of port/APS events (port down, port up, switchover, etc.), received from a port manager code entity, into trunk events. As used herein, a "port" refers to an interface of the networking device to an external communication media. For example, an OC-48 port may consist of an interface to what is generally referred to as a Synchronous Optical Network (SONET) OC-48 communications channel located on a port interface card (PIC) within the networking device. Further, as used herein, the term "APS group" refers to a set of ports that participate in a particular SONET APS (Automatic Protection Switching) configuration, consisting of one protection port and one or more working ports.

(d) Reporting of relevant trunk events to a routing code entity.

(e) Performing connectivity checks for each trunk.

Further in an illustrative embodiment, the trunk manager program code within the trunk, circuit and port manager functionality 74 runs as a Linux process both on an SMC card, when it executes on an actual networking device, and when it executes as part of a network node executable image executing on an emulator machine. The trunk manager program code may further interface with routing, resource manager, port manager, link manager, and network management software entities through any appropriate messaging protocol

performed using Linux sockets. For example, an interface to a network management software entity may be used to receive trunk configuration commands. Similarly, a trunk manager interface to a routing entity may be used to inform the routing entity of any trunk events that are of interest to the routing entity (e.g., add/delete a trunk). An interface to a routing entity may further be used to reserve bandwidth on a particular trunk when setting up a virtual circuit, and to release bandwidth when a virtual circuit is torn down. A trunk manager interface to a port manager entity may be used to receive port and APS information from the port manager. A trunk manager interface with a link manager entity may be used to learn about the configuration of a remote port so that the trunk manager may perform connectivity checks.

Also in the trunk, circuit and port manager functionality 74, may be included port manager program code embodied as one or more "port manager" program code entities. During operation, each port manager entity may generate a number of port/APS events to the trunk manager. The trunk manager then updates its own database based on such events, translates the port events into trunk events, and reports them to the routing and/or signaling entities. In an illustrative embodiment, there are five types of messages exchanged between the trunk manager entity and the port manager entity: Add, Delete, Update, Hello, and Synch. These messages are now described in further detail.

After a physical port/APS is first configured successfully, a port manager entity sends an Add message to the trunk manager entity. Included in the message are the following parameters:

(a) Port ID. If APS is configured, this is the protection port ID of the APS group. Otherwise, this is the port to be added.

(b) APS configuration information.

(c) Bandwidth. Size of quantum and count.

(d) Administrative information regarding the type of port added.

As a result of the Add command, if APS is configured, one APS object and multiple port objects will be created within the trunk manager entity. Otherwise, only one port object will be created.

After a port/APS is deleted successfully, the port manager entity sends a Delete message to the trunk manager. Included within the Delete message is the following parameter: Port ID. The protection port ID of the APS group, or the port ID of the port, to be deleted from the trunk manager entity's database.

In a first embodiment, if the deleted port is part of an APS group, the entire APS group is deleted. Partial deletion may be provided in an alternative embodiment.

The port manager entity sends an Update message to the trunk manager entity when there is a change in port/APS status. Update messages are further classified into six subtypes: Arch Change, Working-to-Protect Switchover, Protect-to-Working Switchover, Protect Down, Protect Up, and APS Down, which are now further described.

The port manager entity sends an Arch Change message to the trunk manager entity when the APS architecture is changed. Included in the Arch Change message is a parameter indicating the Physical port ID of the protection port in the APS group.

ATTORNEY DOCKET NO. SYCMR-031XX
WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES, LLP
TEL. (617) 542-2290
FAX. (617) 451-0313

The port manager entity sends the Working-to-Protect Switchover message to the trunk manager entity when it has successfully performed a switchover from a working port to the protection port. Included in the message are the following parameters:

(a) Physical port ID of the protection port in the APS group.

(b) Physical port ID of the working port in the APS group.

The port manager entity sends the Protect-to-Working Switchover message to the trunk manager entity when it has successfully performed a switchover from the protection port to a working port. Included in the message are the following parameters:

(a) Port ID of the protection port in the APS group.

(b) Port ID of the working port in the APS group.

The port manager entity sends the Protection Up message to the trunk manager entity when a protection port changes status from down to up. Included in the message is the following parameter: Port ID of the protection port in the APS group.

The port manager entity sends the Protection Down message to the trunk manager when the protection port changes status from up to down. Included in the message is the following parameter: Port ID of the protection port in the APS group.

The port manager entity sends the APS Down message to the trunk manager entity when it determines that an entire APS group is down. This message includes the following parameter: Port ID of the protection port in the APS group.

ATTORNEY DOCKET NO. SYCMR-031XX
WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES, LLP
TEL. (617) 542-2290
FAX. (617) 451-0313

Both the trunk manager and port manager entities use Hello message to inform the other that synchronization is needed. Included within the Hello message are the following parameters:

(a) Sequence Number.

(b) Hello Type. May indicate either "initiating synch" or "acknowledging synch."

The port manager sends Synch messages to the trunk manager in order to download its port database. A Synch message consists of multiple Add messages, each for an APS group. Included in the Synch message are the following parameters:

(a) Sequence Number.

(b) List of Add Messages. Empty list signals the end of synchronization.

In addition to the trunk manager and port manager entities, a circuit manager entity may further be included within the trunk, circuit and port functionality 74 shown in Fig. 4. The circuit manager entity may, for example, maintain the state of various virtual circuits used for communication over associated trunks between actual networking devices. Accordingly, the circuit manager code entity operates to set-up and tear-down various virtual circuits. One or more virtual circuits may, for example, be carried over a single trunk.

The platform code 76 shown in the network node executable image 70 of Fig. 4 is responsible for registering any processes used to provide the various code entities within the network node executable image 70. The platform

ATTORNEY DOCKET NO. SYCMR-031XX
WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES, LLP
TEL. (617) 542-2290
FAX. (617) 451-0313

code may further operate to read any necessary configuration information, such as may be stored within one or more configuration files, that is used to configure the various code entities of the network node executable image. Further, the platform code 76 is operable to maintain one or more Management Information Base (MIB) that may be used by the network management functionality of the device. In addition, the platform functionality may operate to control a network management command line interface to the network device. In these ways, the platform 76 operates to maintain various resources, such as databases, that are employed by various other entities within the networking device for various reasons.

In an illustrative embodiment in which the trunk manager program code, circuit manager program code, port manager program code, and signaling and routing program code are each implemented as one or more Linux processes, the operating systems shown in the emulator machines of Figs. 1-2 would each be the Linux operating system. Similarly, in such an embodiment, the operating system of the SMC on which the trunk manager program code executes within the actual networking device would also be Linux. Such commonality of operating systems between the actual networking devices and the emulator machines advantageously enables the same executable code to used on the emulator machines as on the actual networking devices.

Further in an illustrative embodiment, one or more of the program code entities within the network node executable image may be implemented as what are conventionally referred to as "daemons", which is a term generally used to refer to programs that run continuously and exist for the purpose of

handling periodic service requests. Each such daemon program forwards the service requests it receives to other programs (or processes) as appropriate.

Moreover, while the description of the trunk and port manager entities above has been with reference to actual trunks and ports, the trunk and port manager entities maintain and operate in conjunction with emulated trunks and ports when operating within one or more emulator machines. The virtual circuit manager entity within the network node executable image is also operable to support both emulated and actual virtual circuits, depending on whether it is executing within an emulator machine or a physical networking device, respectively. Similarly, the routing and signaling entities are capable of maintaining and operating in conjunction with a topology of an emulated network, just as they would maintain and operate in conjunction with a topology of an actual network topology when executing in a physical networking device.

Fig. 5 illustrates how two network node executable images operate, both when loaded within respective physical networking devices, and when loaded onto one or more emulator machines. As shown in Fig. 5, the network node executable image 100 associated with a Networking Device A includes a signaling and routing functionality 104, which exchanges control information 106 with the signaling and routing functionality 108 within the network node executable image 102 of a Networking Device B 102. The communications between a physical Networking Device A and a physical Networking Device B are provided over a fiber optic cable 116. The network node executable images 100 and 102 for Networking Device A and Networking Device B each include

Internet Protocol (IP) tunnel supporting program code, shown as IP Tunnel code 110 and IP Tunnel code 112. An IP Tunnel is thus provided for communication between the network node executable image 100 and the network node executable image 102. In the case where the two network node executable images of Fig. 5 are operating on one or more emulator machines, then the IP Tunnel is provided over an Ethernet local area network 114. Alternatively, when the two network node executable images of Fig. 5 are operating within respective physical networking devices, the IP Tunnel is provided over the fiber optic cable 116.

In an illustrative embodiment, each of the network node executable images 100 and 102 of Fig. 5 include a code component used to provide communication during emulator operation, and an alternative code component used during operation within an actual physical networking device. In this illustrative embodiment, these alternative code components are provided by alternative network layer drivers, also referred to as protocol driver images, each of which may provide the IP Tunnel functionality shown by IP Tunnel code 110 and IP Tunnel code 112 in Fig. 5. The appropriate network layer driver is loaded for execution into each network node executable image based on configuration information indicating whether the specific network node executable image to be used within an emulator machine, or within an actual physical networking device.

Fig. 6 shows an illustrative embodiment of a physical networking device on which the disclosed network node executable image may be executed. As shown in Fig. 6, a card cage 120 is provided that can receive a number of cards, including a Switch Management Card (SMC) 122, Port

ATTORNEY DOCKET NO. SYCMR-031XX
WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES, LLP
TEL. (617) 542-2290
FAX. (617) 451-0313

Interface Cards (PICs) 124 an 126, and a number of other cards 128. As used herein the term "card" is used to refer to a printed circuit board that provides a number of predetermined functions and/or services. The PICs 124 and 126 each include a number of interfaces to communication media external to the physical networking device. The other cards 128 may, for example, include a switch fabric used to direct data units received on the communication media interfaces of the PICs 124 and 126 to outgoing interfaces. The SMC 122 may include a microprocessor and associated memory for executing a network node executable image, such as the network node executable image 70 shown in Fig. 4. Accordingly, the SMC 122 may include an operating system equivalent to that used on the emulator machine or machines on which the network node executable image is executed in an emulation environment. In an illustrative embodiment, at least a portion of the executable code for the cards other than the SMC 122 in the physical networking device of Fig. 5 is generated from a set of source code which may also be used to generate executable code that may be used in conjunction with, or as part of, the network node executable image 70 shown in Fig. 4.

Those skilled in the art should readily appreciate that the programs defining the functions of the present invention can be delivered to a computer in many forms, including, but not limited to: (a) information permanently stored on non-writable storage media (e.g. read only memory devices within a computer such as ROM or CD-ROM disks readable by a computer I/O attachment); (b) information alterably stored on writable storage media (e.g. floppy disks and hard drives); or (c) information conveyed to a computer through

communication media for example using baseband signaling or broadband signaling techniques, including carrier wave signaling techniques, such as over computer or telephone networks via a modem. In addition, while the invention may be embodied in computer software, the functions necessary to implement the invention may alternatively be embodied in part or in whole using hardware components such as Application Specific Integrated Circuits (ASICs), Programmable Logic Arrays (PLAs), or other hardware, or some combination of hardware components and software.

While the invention is described through the above exemplary embodiments, it will be understood by those of ordinary skill in the art that modification to and variation of the illustrated embodiments may be made without departing from the inventive concepts herein disclosed. Accordingly, the invention should not be viewed as limited except by the scope and spirit of the appended claims.

CLAIMS

What is claimed is:

1. A network emulator for providing an emulated network, said emulated network including a plurality of emulated networking devices, comprising:

a plurality of network node executable images, each one of said network nodes corresponding to one of said plurality of emulated networking devices; and

wherein each of said network node executable images includes executable code that is operable to execute without modification on a physical networking device corresponding to one of said plurality of emulated networking devices, and wherein said executable code within each of said network node executable images that is operable to execute without modification on a corresponding physical networking device includes a program component operable to perform signaling and routing.

2. The network emulator of claim 1, wherein said executable code that is operable to execute without modification on said one of said plurality of emulated networking devices is operable to be executed within one of several cards that make up said one of said plurality of emulated networking devices.

3. The network emulator of claim 1, wherein said executable code that is operable to execute without modification on said one of said plurality of emulated networking devices includes a program component operable to maintain a representation of a topology of said emulated network.

ATTORNEY DOCKET NO. SYCMR-031XX
WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES, LLP
TEL. (617) 542-2290
FAX. (617) 451-0313

4. The network emulator of claim 1, wherein said executable code that is operable to execute without modification on said one of said plurality of emulated networking devices includes a program component operable to maintain at least one emulated trunk between a corresponding one of said plurality of emulated networking devices and an adjacent one of said plurality of emulated networking devices.

5. The network emulator of claim 1, wherein said executable code that is operable to execute without modification on said one of said plurality of emulated networking devices includes a program component operable to maintain at least one emulated virtual circuit between a corresponding one of said plurality of emulated networking devices and another one of said plurality of emulated networking devices.

6. The network emulator of claim 1, wherein a plurality of said network node executable images execute on a single emulator machine, wherein said emulator machine includes at least one processor and at least one memory.

7. The network emulator of claim 1, wherein each of said plurality of network node executable images executes on a respective one of a plurality of emulator machines, each one of said plurality of emulator machines including at least one processor and at least one memory.

8. The network emulator of claim 7, wherein said plurality of emulator machines are interconnected by a local area network.

ATTORNEY DOCKET NO. SYCMR-031XX
WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES, LLP
TEL. (617) 542-2290
FAX. (617) 451-0313

9. The network emulator of claim 8, wherein said local area network is an Ethernet local area network.

10. The network emulator of claim 1, wherein at least one of said plurality of network node executable images executes on an associated plurality of emulator machines.

11. The network emulator of claim 1, wherein each of said plurality of network node executable images further comprises:

a first protocol driver image operable to provide communications between said corresponding one of said plurality of emulated networking devices and other ones of said plurality of emulated networking devices within said network emulator; and

a second protocol driver image operable to provide communications between a physical networking device of said physical networking device type associated with said corresponding one of said plurality of emulated networking devices and other physical networking devices.

12. A method of providing an emulated network, said emulated network including a plurality of emulated networking devices, comprising:

loading a plurality of network node executable images onto at least one emulator machine, each one of said network nodes corresponding to one of said plurality of emulated networking devices; and

wherein each of said network node executable images includes executable code that is operable to execute without

modification on a physical networking device corresponding to one of said plurality of emulated networking devices, and wherein said executable code within each of said network node executable images that is operable to execute without modification on a corresponding physical networking device includes a program component operable to perform signaling and routing.

13. The method of claim 12, wherein said executable code that is operable to execute without modification on said one of said plurality of emulated networking devices is operable to be executed within one of several cards that make up said one of said plurality of emulated networking devices.

14. The method of claim 12, further comprising maintaining, by a program component within said executable code that is operable to execute without modification on said one of said plurality of emulated networking devices, a representation of a topology of said emulated network.

15. The method of claim 12, further comprising maintaining, by a program component within said executable code that is operable to execute without modification on said one of said plurality of emulated networking devices, at least one emulated trunk between a corresponding one of said plurality of emulated networking devices and an adjacent one of said plurality of emulated networking devices.

16. The method of claim 12, further comprising maintaining, by a program component within said executable code that is operable to execute without modification on said one of said

plurality of emulated networking devices, at least one emulated circuit between a corresponding one of said plurality of emulated networking devices and another one of said plurality of emulated networking devices.

17. The method of claim 12, further comprising executing a plurality of said network node executable images execute on a single one of said at least one emulator machine, wherein said single emulator machine includes at least one processor and at least one memory.

18. The method of claim 12, further comprising executing each of said plurality of network node executable images executes on a corresponding one of said at least one emulator machine, said corresponding one of said at least one emulator machine including at least one processor and at least one memory.

19. The method of claim 18, further comprising communicating among said plurality of emulator machines over a local area network.

20. The method of claim 19, wherein said communicating comprises communicating over an Ethernet local area network.

21. The method of claim 12, further comprising executing at least one of said plurality of network node executable images executes on an associated plurality of said at least one emulator machine.

ABSTRACT OF THE DISCLOSURE

A system and a method for providing an emulated network including a plurality of emulated networking devices, in which a number of network node executable images are employed. Each of the network node executable images corresponds to one of the emulated networking devices. The emulated networking devices represent physical networking devices, such as optical switches. A portion of each of the network node executable images is operable to execute without modification on a physical networking device. Each of the network node executable images includes a number of program components, such as a topology related program component for maintaining a representation of the topology of the emulated network, a trunk manager program component for maintaining at least one emulated trunk between the corresponding emulated networking device and an adjacent other emulated networking device, and a circuit manager program component operable to maintain at least one emulated circuit between the corresponding emulated networking device and another one of the emulated networking devices. Various numbers of network node executable images may be executed on a single emulator machine, and multiple emulator machines may be employed to execute a single network node executable image.

233774

ATTORNEY DOCKET NO. SYCMR-031XX
WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES, LLP
TEL. (617) 542-2290
FAX. (617) 451-0313

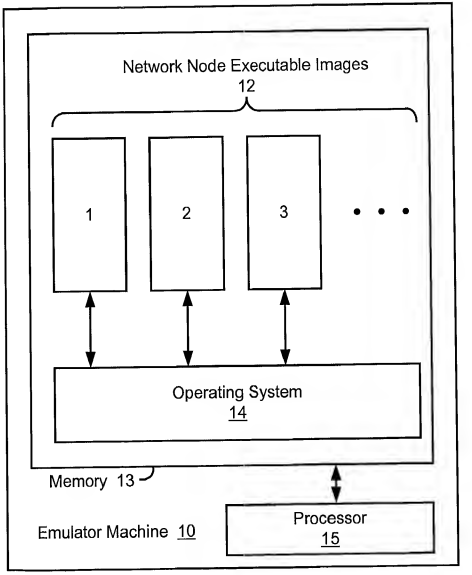
Network Node Executable Images
12
1
2
3
Operating System
14
Memory 13
Emulator Machine 10
Processor
15

Fig. 1

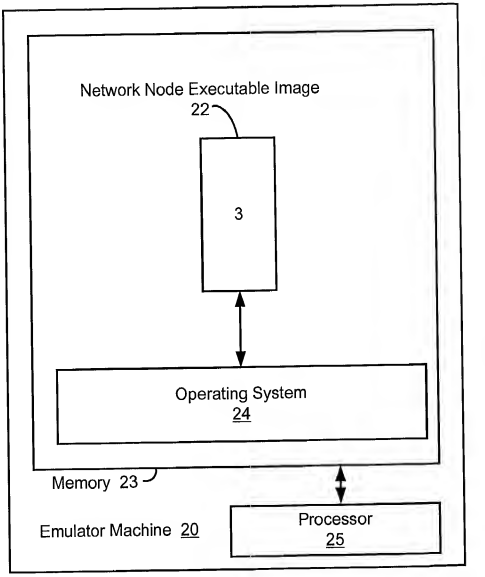
Network Node Executable Image
22
3
Operating System
24
Memory 23
Emulator Machine 20
Processor
25

Fig. 2

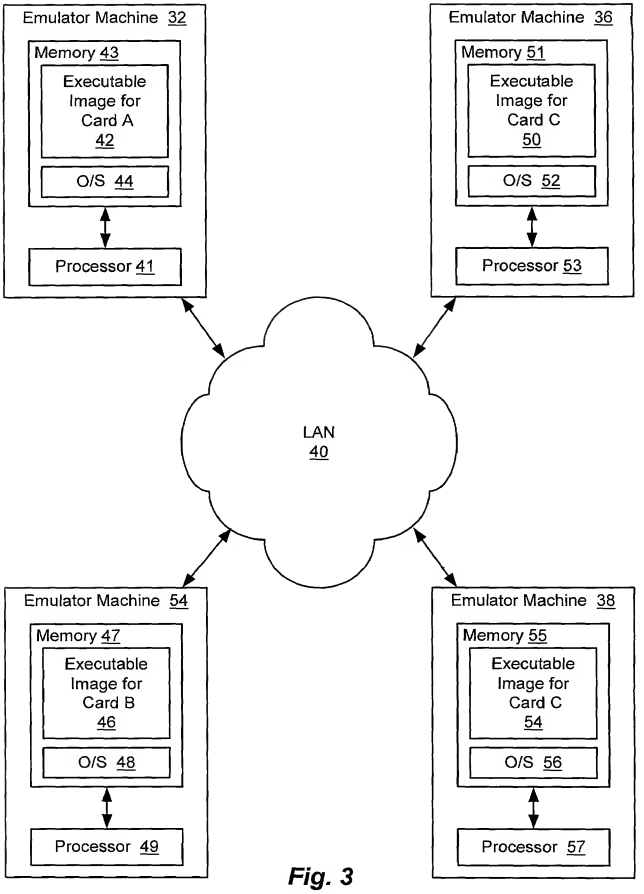
Emulator Machine 32
Memory 43
Executable Image for Card A 42
O/S 44
Processor 41
Emulator Machine 36
Memory 51
Executable Image for Card C 50
O/S 52
Processor 53
LAN 40
Emulator Machine 54
Memory 47
Executable Image for Card B 46
O/S 48
Processor 49
Emulator Machine 38
Memory 55
Executable Image for Card C 54
O/S 56
Processor 57

Fig. 3

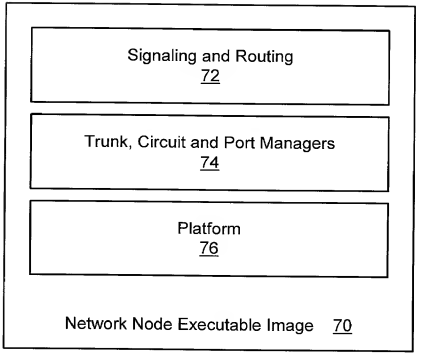
Signaling and Routing
72
Trunk, Circuit and Port Managers
74
Platform
76
Network Node Executable Image 70

Fig. 4

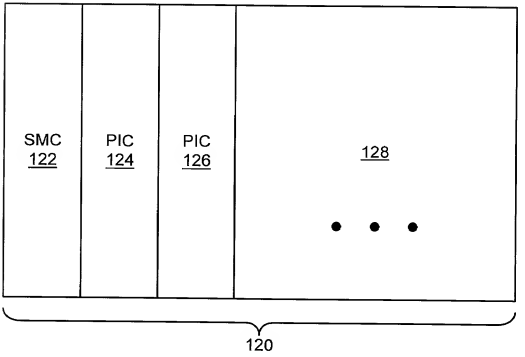
SMC
122
PIC
124
PIC
126
128
120

Fig. 6

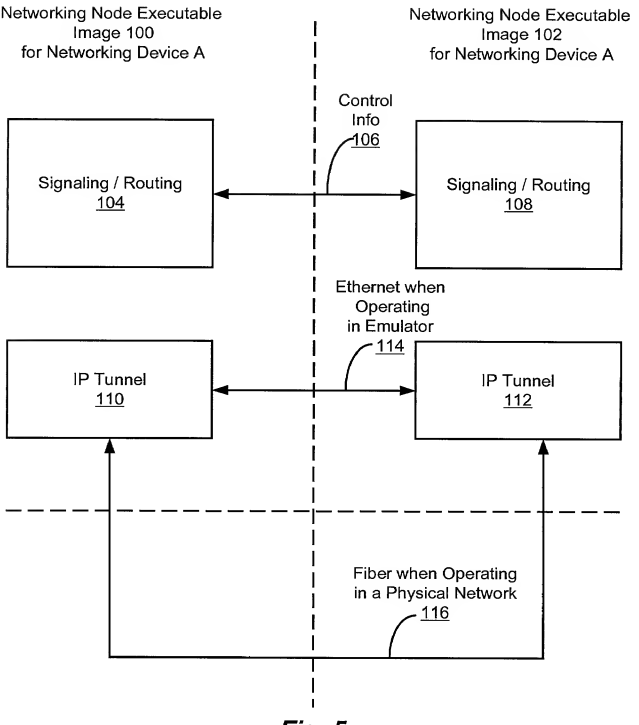
Networking Node Executable Image 100 for Networking Device A
Networking Node Executable Image 102 for Networking Device A
Control Info 106
Signaling / Routing 104
Signaling / Routing 108
Ethernet when Operating in Emulator 114
IP Tunnel 110
IP Tunnel 112
Fiber when Operating in a Physical Network 116

Fig. 5

DECLARATION AND POWER OF ATTORNEY

As a below-named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

NETWORK EMULATOR

the specification of which (check one):

[X] is attached hereto. [ ] was filed _____ as Application No._____
amended on _____(if applicable).

[ ] was filed as PCT International Application No. _____ on _____,
and was amended under PCT Article 19 on _____ (if applicable).

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations §1.56(a).

I hereby claim foreign priority benefits under Title 35, USC §119(a)-(d) of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)		Date Filed	Priority Claimed	
(Number)	(Country)	(Day/Month/Year)	[ ] Yes	[ ] No
(Number)	(Country)	(Day/Month/Year)	[ ] Yes	[ ] No

I hereby claim the benefit under Title 35, USC §119(e) of any United States provisional application(s) listed below:

(Application Number)	(Filing Date)
(Application Number)	(Filing Date)
(Application Number)	(Filing Date)

Express Mail Numb
EL418427798US

Attorney
Docket No.: SYCMR-031XX

I hereby claim the benefit under Title 35 USC §120 of any United States application(s) listed below and insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35 USC §112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Application No.)	(Filing Date)	(Patented/pending/abandoned)
(Application No.)	(Filing Date)	(Patented/pending/abandoned)

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) to prosecute this application and transact all business connected therewith in the Patent and Trademark Office, and to file with the USRO any International Application based thereon.

Stanley M. Schurgin, Reg. No. 20,979
Charles L. Gagnebin III, Reg. No. 25,467
Paul J. Hayes, Reg. No. 28,307
Victor B. Lebovici, Reg. No. 30,864
Eugene A. Feher, Reg. No. 33,171
Beverly E. Hjorth, Reg. No. 32,033
Holliday C. Heine, Reg. No. 34,346
Gordon R. Moriarty, Reg. No. 38,973

Address all correspondence to:

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
Ten Post Office Square
Boston, Massachusetts 02109
Telephone: (617) 542-2290
Telecopier: (617) 451-0313

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Sole/First Inventor: Chikong Shue		
City of Residence Andover	State or Country Massachusetts	Country of Citizenship USA
Post Office Address 2 Possum Hollow Road	City Andover	State or Country Zip Code Massachusetts 01810
Signature: (Please sign and date in permanent ink.) X		Date signed: X

Attorney
Docket No.: SYCMR-031XX

Name of Second Joint Inventor: John Thomas Moy		
City of Residence Acton	State or Country Massachusetts	Country of Citizenship USA
Post Office Address 8 Castle Drive	City Acton	State or Country Zip Code Massachusetts 01720
Signature: (Please sign and date in permanent ink.) X		Date signed: X

Full Name of Third Joint Inventor: Tao Yang		
City of Residence Acton	State or Country Massachusetts	Country of Citizenship Canada
Post Office Address 22 Lexington Drive	City Acton	State or Country Zip Code Massachusetts 01720
Signature: (Please sign and date in permanent ink.) X		Date signed: X

Full Name of Fourth Joint Inventor: Raymond Xie		
City of Residence Westford	State or Country Massachusetts	Country of Citizenship Canada
Post Office Address 2 True Bean Way	City Westford	State or Country Zip Code Massachusetts 01720
Signature: (Please sign and date in permanent ink.) X		Date signed: X